پردازش گراف های داده ای به وسیله مدل های وفقی و کاربرد آن در پردازش تصاویر

زینب رضایی*
گروه کامپیوتر دانشگاه آزاد اسلامی واحد اردبیل

چکیده

شبکه های عصبی بازگشت پذیر یکی از مدل های وفقی اند که برای پردازش گراف ها مناسبند. مدل ابتدایی این شبکه ها تنها قادر به پردازش گراف های بدون حلقه مرتب شده جهت دار ($DAOG$) هستند. در این گراف ها فرزندان هر نود مرتب شده اند، این فرض یک محدودیت غیر ضروری است. در این مقاله روش اشتراک گذاری وزنی که محدودیت مرتب بودن فرزندان را درپردازش گراف ها رفع می کند، معرفی شده است. همچنین برای پردازش گراف های بدون چرخه جهت دار با لبه های بر چسب خورده ($DAG-LE$) نیز مدل جدیدی از شبکه های عصبی بازگشت پذیر معرفی می شود. هر دو روش، استقلال خروجی شبکه نسبت به جایگشت های لبه های خارج شده از هر نود را تضمین می کنند. در ادامه یک تشخیص دهنده الگو بر پایه مدل بازگشت پذیراخیر آمده است. تشخیص با این روش نیاز به هیچ آگاهی قبلی از مدل شی مورد نظر ندارد ودر مقابل انتقال وچرخش تصویر نیز پایا است.

کلمات کلیدی: شبکه های عصبی بازگشت پذیر، پردازش گراف ها، اشتراک گذاری وزنی، تشخیص دهنده الگو، نمایش گرافیکی.

۱ مقدمه

در اکثر کاربردها اطلاعاتی که برای حل مسایل مطرح می شود در قالب عناصر و روابط بین آن ها سازماندهی شده اند. ساده ترین نوع داده دینامیکی دنباله ها هستند که برای نمایش وابستگی زمانی به کار می روند و توسط شبکه های عصبی بازگشتی[1] پردازش می شوند. این شبکه ها ابزار قوی در مدل سازی پدیده های وابسته به زمان هستند، اطلاعات نمادی و زیر نمادی موجود در دنباله ها به وسیله مدل های وفقی پردازش می شوند. اما غالباً در اغلب کاربردها از ساختارهای پیچیده تر مثل گراف ها و درخت ها استفاده می شود.

* عهده‌دار مکاتبات(آدرس پست الکترونیکی: rezaii57@yahoo.com)

[1] Recurrent neural network

شبکه های عصبی بازگشت پذیر[۲] مدل جدید، مناسبی برای پردازش گراف ها هستند که به عنوان توسیع شبکه های باز گشتی در نظر گرفته می شوند.

برطبق مدلی که در [۵] پیشنهاد شده است شبکه های عصبی بازگشت پذیر تنها با خانواده محدودی از گراف ها، $DOAG$[۳]ها و $DPAG$[۴] ها کار می کنند [۹و۱۵]. در $DPAG$ هر لبه خارج شده از هر نود موقعیت خاص و تعریف شده ای دارد. به عبارت دیگر هر ترتیب دیگری از بچه ها گراف متفاوتی را ایجاد می کند. این فرض در اغلب کاربردها مثلا درنمایش بسیاری از ترکیبات شیمیایی و مدل سازی الگو یک محدودیت غیر ضروری می باشد. مساله تغییر ناپذیری یا تقارن در خیلی از حوزه ها مثل فیزیک، ریاضی، مهندسی و.... وجود دارد،کاربردهای مهم آن در بینایی ماشین و در نظریه سیگنال ها و تشخیص الگو در منابع [۱۱ و ۱۲] آمده است.

در گرایش شبکه های عصبی کوشش های زیادی در ساخت معماری های خاص کاربردی پایا در مقابل جایگشت های ورودی ها انجام شده است،که اغلب این پایایی بر پایه بعضی پیش پردازش ها یا انتخاب صفاتی که در مقابل انتقال ورودی پایا هستند، انجام می گیرد، و در بعضی حالات خود مدل های شبکه ای، پایایی خروجی را تحت انتقال های ورودی خاص حفظ می کنند [۱۳]. در این مقاله روش اشتراک گذاری وزنی[۵] برای رفع محدودیت روی مرتب سازی در DAG[۶]ها و روشی خاص برای پردازش $DAG-LE$[۷]ها آمده است[۱و۶]. هر دو روش، استقلال خروجی شبکه را در مقابل جایگشت های لبه های خارج شده از هر نود حفظ می کنند. قابل توجه است که روش اشتراک گذاری وزنی تنها در گراف هایی با اتصالات پایین کاربرد دارد، چرا که تعداد نرون ها به صورت فاکتوریلی از بالا ترین درجه خروجی رشد می کند، اما روش ارایه شده برای $DAG-LE$ها این محدودیت را نیز ندارد.

۲ معرفی نمادها گراف جهت دار $G(V,A,\ell)$ را که V مجموعه ای از نودها، $A\subseteq V^2$ مجموعه لبه ها و $\ell:V\longrightarrow L$ تابع برچسب گذاری و $L\subseteq R^m$ مجموعه متناهی از بر چسب ها است در نظر می گیریم. برای هر نود $v\in V$، نماد $ch[v]$ مجموعه فرزندان v را نشان می دهد. درجه خروجی هر نود v که با $od[v]$ نشان داده می شود کاردینال مجموعه $ch[v]$ است و $o=\max_{v\in V}\{od[v]\}$ بیشترین درجه خروجی است. هر لبه (v,w) در گراف بر چسب خورده نشان دهنده ارتباط علّی بین متغییر های موجود در v و w است.

DAG کلاسی از گراف ها است که هیچ مسیری از هر نود به خودش ندارد و لبه ها جهت دار هستند.(یعنی $(v,w)\neq(w,v)$) بنابراین روی مجموعه نودهای یک DAG یک رابطه تر تیبی جزئی تعریف می شود به طوری که $v\leq w$ اگر مسیری از v به w وجود داشته باشد. $DPAG$ها از DAG متفاوت هستند چراکه ترتیب برای فرزندان هر نود به وسیله تابع یک به یک $o_v:ch[v]\longrightarrow\{1,2,\ldots,o\}$ تعریف شده که مکان $o_v(c)$ به هر فرزند c از نود v داده می شود. مکان هر فرزند صفت شاخص برای $DPAG$ها است چرا که دو

[2] Recursive neural network (RNN)
[3] Directed ordered acyclic graph
[4] Directed positional acyclic graph
[5] Weight sharing
[6] Directed acyclic graph
[7] Directed acyclic graph with labeled edges

$DPAG$ مساوی هستند اگر و فقط اگر تناظر یک به یک نه تنها بین نود ها و لبه ها بلکه بین فرزندان هر نود وجود داشته باشد. بنابراین، از دیدگاه نمادی $DPAG$ ها توسط چهار تایی (V, A, ℓ, O) نمایش داده می شوند که $O = \{o_1, \ldots, o_{|V|}\}$ مجموعه ای از توابع تعیین کننده مکان فرزندان است. در DAG ها فرزندان هر نود v دقیقا توسط مجموعه $ch[v]$ مشخص می شوند که برای $DPAG$ ها فرزندان می توانند به طور خاص توسط برداری با بعد ثابت $[v_1, \ldots, v_o]$ نمایش داده شوند. اگر فرزندی برای مکانی وجود نداشته باشد، آن جزء null درنظرگرفته می شود. سرانجام لازم به ذکر است که تمام گراف ها منبع دارند. منبع s نودی است که از تمام نودهای دیگرقابل دسترسی باشد ($s \le v$ به ازای هر $s \neq v \in V$).

۳ شبکه های عصبی بازگشت پذیر برای پردازش $DPAG$ ها

شبکه ها عصبی بازگشت پذیر یک $DPAG$ را بر طبق نمای توصیف شده در شکل (۱) پردازش می کنند. شبکه عصبی بازگشت پذیر بر طبق ساختار گراف گسترش می یابد و شبکه کد شده را ایجاد می کند. در هر نود v حالت X_v با تابع انتقال از برچسب U_v و حالت فرزندانش $X_{v_o}, \ldots, X_{v_1}$ محاسبه می شود

$$X_v = f(X_{v_1}, \ldots, X_{v_o}, U_v, \theta_f).$$

θ_f برداری از پارامترهای مستقل از نود v می باشد. به بیان دیگر حالت هر نود با استراتژی پایین به بالا محاسبه می شود، یعنی X_v بعد از اینکه حالت تمام فرزندان v موجود باشد محاسبه می شود. اگر i ـ امین فرزند موجود نباشد ($v_i = null$)، X_{v_i} با حالت X_{null} از پیش تعریف شده مقدار دهی می شود. در منبع یک تابع خروجی به وسیله شبکه پیشرو که شبکه خروجی نامیده می شود، محاسبه می شود

$$y_s = g(X_s, \theta_g)$$

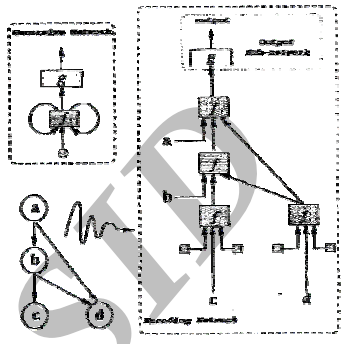
نمایش پارامتری g, f می تواند به وسیله مدل های مختلف شبکه های عصبی پیاده سازی شود. در حالت پرسپترون سه لایه N، با توابع فعالیت سیگموید در واحدهای مخفی و توابع فعالیت خطی در واحدهای خروجی، f تابع ورودی ـ خروجی f_N است. که درآن وابستگی نود v به برچسب فرزندانش با رابطه زیر نمایش داده می شود:

$$\begin{aligned} X_v &= f_N(X_{v_1}, \ldots, X_{v_o}, U_v, \theta_f) \\ &= V.\vec{\sigma}\left(\sum_{k=1}^{o} A_k.X_{v_k} + B.U_v + C\right) + D \end{aligned} \qquad (۱)$$

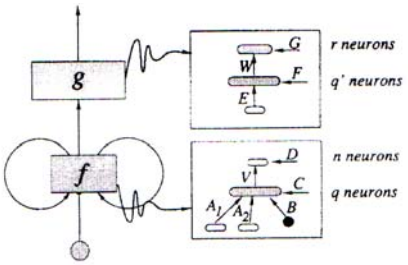
که در آن $\vec{\sigma}$ تابع سیگموید برداری و θ_f (وزن های شبکه)، $A_k \in R^{q,n}$ ، $k = 1, \cdots, o$ ، $B \in R^{q,m}$ ، $C \in R^q$ ، $D \in R^n$ و $V \in R^{n,q}$ را جمع می کند. در اینجا m بعد فضای بر چسب، n بعد فضای حالت و q تعداد نرون های مخفی را نشان می دهد. معادله مشابه برای g نیز برقرار است

$$y_s = g_N(X_s, \theta_g) = W\vec{\sigma}(E.X_s + F) + G$$

که در آن θ_g جمع آوری کننده $E \in R^{q',n}$، $F \in R^{q'}$، $G \in R^{r}$ و $W \in R^{r,q'}$ شکل (۲). بنابراین شبکه عصبی بازگشت پذیر، تابع $h : DPAG_s \to R^r$ را که $h(G) = Y_s$ پیاده سازی می کند. در واقع $h = go\overline{f}$ که درآن $\overline{f}(G) = X_s$ نشان دهنده پردازشی است که یک گراف را گرفته و حالت آن را در منبع برمی گرداند [۷ و ۸ و ۴].



شکل ۱: شبکه های کد شده و خروجی نسبت داده شده به $DPAG$. شبکه عصبی بازگشت پذیر بر طبق ساختار گسترش می یابد.



شکل ۲: پیاده سازی پرسپترون سه لایه معماری نشان داده شده در شکل ۱ در هر دو شبکه پیشرو لایه های مخفی سیگموید و لایه های خروجی خطی هستند.

٤ شبکه های عصبی باز گشت پذیر برای پردازش DAG ها

شبکه های عصبی باز گشت پذیر برای کار مستقیم با $DPAG$ها طراحی شده اند. تابع f به طور طبیعی مکان هر فرزند را در یک نود در نظر می گیرد، چرا که حالت هر فرزند مکان خاصی در تابع ورودی f دارد. روش ساده برای پردازش DAG ها به این صورت است که هر $DAG(V, A, \ell)$ ورودی، به $DPAG(V, A, \ell, O)$ نگاشته شود، که به جای DAG اصلی پردازش می شود. این انتقال ما را ملزم به تخصیص مکان به فرزندان هر نود می کند. با این وجود این کار نتیجه محاسبات را تحت تاثیر قرار می دهد. برای اجتناب از این وضعیت f باید علیرغم مکان نسبت داده شده به فرزندان نتیجه یکسانی را تولید کند. در واقع f باید در رابطهٔ

$$f(X_1, \ldots, X_o, U, \theta_f) = f(X_{\pi(1)}, \cdots X_{\pi(o)}, U, \theta_f) \qquad (۲)$$

برای هر جایگشت π هر $X_1, \cdots, X_o \in R^n$ ، $U \in R^m$، θ_f صدق کند.

توجه می کنیم که در انتقال هر $DAG(V, A, \ell)$ به تمام $DPAG_s(V, A, \ell, O_1)$، ...، $DPAG_s(V, A, L, O_d)$ که تنها در ترتیب فرزندان متفاوتند، الگوریتم آموزش باید تابع f را که در (۲) صدق کند تولید نماید. با وجود این در عمل این کار نیاز به یک مجموعه عظیم آموزشی و زمان طولانی آموزش دارد. به روش دیگر می توان

f را به وسیله مدلی که به طور طبیعی در (۲) صدق می کند پیاده سازی کرد، یعنی مدلی که بتواند تمام توابع تعریف شده در (۲) را پیاده سازی کند. در ادامه نشان داده شده است که چگونه شبکه عصبی سه لایه با قیود مناسب می تواند برای این هدف استفاده شود.

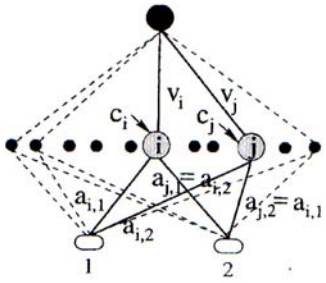
۵ یک معماری شبکه برای پردازش DAG ها

این مدل شبیه مدل مطرح شده توسط $LeCun$ در [۱۴] است. به طور مثال شبکه عصبی سه لایه با دو واحد ورودی، $2q$ واحد مخفی ویک واحد خروجی را در نظر می گیریم (شکل ۳). iـ امین نرون مخفی با خروجی شبکه توسط مقدار $v_i\sigma\left(a_{i,1}x_1+a_{i,2}x_2+c_i\right)$ مربوط می شود که در آن x_1و x_2 ورودی اند و $a_{i,1}$، $a_{i,2}$، c_i و v_i پارامترهای شبکه اند. فرض کنیم برای j ـ امین واحد مخفی وزن های پایه مخفی به ورودی تعویض می شوند ($a_{j,2}=a_{i,1}$ و $a_{j,1}=a_{i,2}$) که سایر وزن ها یکسان هستند ($v_j=v_i$ و $c_j=c_i$). بنابراین سهم i، j در خروجی شبکه به صورت

$$h_i\left(x_1,x_2\right)=v_i\sigma\left(a_{i,1}x_1+a_{i,2}x_2+c_i\right)+v_i\sigma\left(a_{i,2}x_1+a_{i,1}x_2+c_i\right)$$

است که $h_i\left(x_1,x_2\right)=h_i\left(x_2,x_1\right)$ که حالت خاصی از محدودیت (۲) است. اگر این رفتار توسط q زوج از واحدهای مخفی انتخاب شده خاص به اشتراک گذاشته شود، تمام خروجی شبکه می تواند توسط

$$f\left(x_1,x_2\right)=\tfrac{1}{2}\sum_{i=1}^{2q}h_i\left(x_1,x_2\right) \qquad (۳)$$



شکل ۳: شبکهٔعصبی با وزن های مشترک

محاسبه شود که f محدودیت (۲) را کاملا محقق می سازد.

مباحث مثال بالا می تواند به راحتی برای معماری های عمومی تر مثل شبکه های با ورودی و خروجی های زیاد به کار برده شود. در واقع برای پیاده سازی تابع $f\left(X_1,\cdots,X_n\right)$ که نسبت به جایگشت های ورودی پایا باشد، لایه مخفی شبکه باید شامل مجموعه واحدهایی باشد که وزن های ورودی به مخفی را به اشتراک می گذرانند. هر مجموعه باید شامل واحدی برای هر جایگشت از وزن ها باشد، به این معنی که یک نورون i با وزن های $\left(a_{i_1},a_{i_2},a_{i_3}\right)$، یک نورون j با وزن های $\left(a_{j_1}=a_{i_2},a_{j_2}=a_{i_1},a_{j_3}=a_{i_3},\ldots\right)$، یک نورون k

باوزن های $\left(a_{k_1}=a_{i_1},a_{k_2}=a_{i_3},a_{k_3}=a_{i_2},\cdots\right)$ وغیره. به علاوه (۲)یک محدودیت متفاوت در مورد عدم حساسیت به جایگشت های تمام ورودی ها فراهم می آورد، چرا که f باید بدون تغییر بماند مگر زمانی که حساسیت X_{v_k} عوض شود. با این همه، توضیح زیر برای ساختن شبکه ای که در (۲) صدق کند، اساسا چیزی است که تا کنون بیان شده است. فرض کنیم $P=\left\{\pi_1,\cdots,\pi_p\right\}$، مجموعه تمام جایگشت ها روی $\{1,\ldots,o\}$ باشد. شبکه SN را که دارای همان پارامترهای $\left(\theta_f=\left(A_1,\cdots,A_o,B,C,D,V\right)\right)$ شبکه N در (۱)بوده ولی پارامترها در میان تعدادی از اتصال ها به اشتراک گذاشته شده اند، در نظر می گیریم. در واقع این شبکه دارای pq واحد مخفی است. وزن های اتصال مخفی به خروجی $\overline{V}\in R^{n,qp}$، وزن های اتصالات ورودی به مخفی $\overline{A}\in R^{qp,no+m}$، آستانه های مخفی $\overline{C}\in R^{qp}$ و آستانه خروجی $\overline{D}\in R^{n}$ هستند که

$$\overline{V}=\left(V,\ldots,V\right)$$
$$\overline{A}=\left(\begin{array}{ccc|c} A_{\pi_1(1)} & \cdots & A_{\pi_1(0)} & B \\ \vdots & & \vdots & \vdots \\ A_{\pi_p(1)} & \cdots & A_{\pi_p(0)} & B \end{array}\right) \qquad (۴)$$
$$\overline{C}=\left(C,\ldots,C\right)'$$
$$\overline{D}=pD$$

به علاوه فرض کنیم $I_v=\left(X'_{v_1},\ldots,X'_{v_o},U'_v\right)'$ که « ' » اپراتور ترانهاده است. به طور شهودی برای هر واحد مخفی N تعداد p واحد در SN وجود دارند که پارامترهای یکسان را به اشتراک می گذراند. به راحتی دیده می شود که خروجی SN به صورت زیر است:

$$\begin{aligned} f_{SN}\left(X_{v_1},\ldots,X_{v_o},U_v,\theta_f\right)&=\overline{V}.\overline{\sigma}\left(\overline{A}.I_v+\overline{C}\right)+\overline{D} \\ &=\sum_{i=1}^{p}\left[V.\vec{\sigma}\left(\sum_{k=1}^{o}A_{\pi_i(k)}.X_{v_k}+B.U_v+C\right)+D\right] \qquad (۵) \\ &=\sum_{i=1}^{p}f_N\left(X_{\pi_i(v_1)},\ldots,X_{\pi_i(v_o)},U_v,\theta_f\right). \end{aligned}$$

آموزش در شبکه های عصبی بازگشت پذیر با وزن های مشترک می تواند از طریق پس انتشار استاندارد در طول ساختار [۱۰و ۹] همانطورکه در معماری استاندارد انجام می گیرد، حاصل شود. قید تساوی در طول آموزش بسادگی به دلیل شرط تساوی اولیه تضمین می شود. در حقیقت وزن های مخفی به خروجی با همان کیفیت به روز شده و تساوی سهم خطای پس انتشار در وزن های ورودی به مخفی را نیز نتیجه می دهند.

٦ شبکه های عصبی باز گشت پذیر برای پردازش $DAG-LE$

روش اشتراک گذاری وزنی معرفی شده در بالا تنها برای کار با DAG ها با درجه خروجی پایین مناسب است، چرا که پارامترهای شبکه به صورت فاکتوریلی از درجه خروجی رشد می کنند. با هرس کردن اتصالات دارای اطلاعات کم می توان درجه خروجی را کاهش داد. البته با این روش ممکن است بعضی اطلاعات مهم حذف شود. در مدلی که در ادامه معرفی می شود، هر دو محدودیت مرتب بودن و درجه خروجی برای $DAG-LE$ ها مرتفع می شود. در حقیقت برای تابع فعالیت f در $DAG-LE$ ها پارامتری که وابسته به ترتیب باشد تعریف نشده است. مشارکت فرزندان هر نود مقدار $\overline{X}(ch[v])\in R^p$ است که به صورت زیر محاسبه می شود:

$$\overline{X}(ch[v])=\frac{1}{|ch[v]|}\left(\sum_{i=1}^{|ch[v]|}\left(\sum_{j=1}^{k}H_j L^{(j)}_{(v,ch_i[v])}\right)X_{ch_i[v]}\right), \qquad (۶)$$

به طوری که $L_{(v,ch_i[v])}\in R^k$ برچسب متصل به لبه $(v,ch_i[v])$ و $H\in R^{p,n,k}$ ماتریس وزنی است. به ویژه $H_j\in R^{p,n}$، j ـ امین لایه ماتریس H و $L^{(j)}_{(v,ch_i[v])}$، j ـ امین جزء برچسب لبه است. نهایتا حالت، هر نود v به وسیله پرسپترون دو لایه با خروجی خطی به صورت

$$X_v=\overline{f}\left(\overline{X}\left(ch[v]\right),U_v,\theta_{\overline{f}}\right)=V\vec{\sigma}\left(A\overline{X}\left(ch[v]\right)+BU_v+C\right)+D \qquad (۷)$$

است. که در آن $\theta_{\overline{f}}$ جمع کننده $A\in R^{q,p}$، $B\in R^{q,m}$، $C\in R^q$، $D\in R^n$ و $V\in R^{n,q}$ که q تعداد واحدهای مخفی است. در منبع، نیز تابع خروجی به وسیله شبکه پیشرو،

$$Y_s=g(X_s,\theta_s)=W\vec{\sigma}(EX_s+F)+G$$

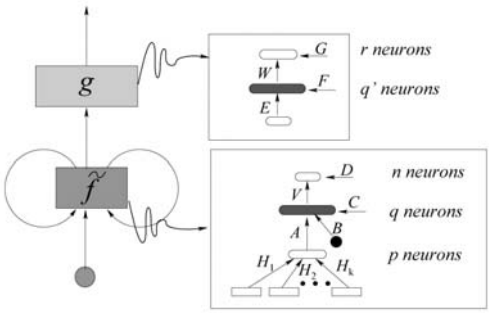
با $E\in R^{q',n}$، $F\in R^{q'}$، $G\in R^r$ و $W\in R^{r,q'}$ محاسبه شده است.

با شروع از معادلات (۶) و (۷) یک پیاده سازی MLP از شبکه بازگشت پذیر می تواند به سادگی با بازنویسی معادله (۶) به شکل

$$\overline{X}(ch[v])=\frac{1}{|ch[v]|}\left(\sum_{i=1}^{k}H_j\left(\sum_{j=1}^{|ch[v]|}L^{(j)}_{(v,ch_i[v])}X_{ch_i[v]}\right)\right) \qquad (۸)$$

حاصل شود.

بنابراین معادله (۸) پیشنهاد می کند که سهم فرزندان هر نود در حالت آن نود با استفاده از پرسپترون سه لایه با k ورودی $\sum_{i=1}^{|ch[v]|}L^{(j)}_{(v,ch_i[v])}X_{ch_i[v]}$، $j=1,2,\ldots,k$ و H_j به عنوان ماتریس وزن ورودی به مخفی قابل محاسبه است. لایه مخفی دوم ولایه خروجی بدون تغییر باقی می مانند وسهم آن ها در محاسبه توسط معادله (۷) توصیف می شود(شکل ۴).



شکل ٤: پیاده سازی MLP برای شبکه بازگشت پذیر. در هر دو شبکهٔ پیشرو لایه های خاکستری سیگموید و لایه های سفید خطی اند.

در بسیاری از کاربردهای پردازش تصویر الگو بطور دقیق تری به وسیله گراف ها ی جهت دار برچسب خورده توصیف می شود. در این بخش یکی از این روش ها ارایه شده است. ابتدا تصویر $DAG-LE$ را بر پایه رنگ بخش بندی می کنیم. الگوریتم استفاده شده K-means clustering می باشد. سپس هر ناحیه به دست آمده از تصویر بخش بندی شده، توسط برداری از صفات با مقادیر حقیقی که اطلاعات هندسی و دیداری ناحیه را جمع می کند، توصیف می شود. به عنوان مثال صفاتی نظیر: میانگین رنگ، مختصات مستطیل احاطه کننده ناحیه، مرکز جرم ناحیه، مساحت ناحیه و زاویه محور اینرسی ناحیه بافق. از طرف دیگر اطلاعات ساختاری مربوط به روابط همسایگی نواحی در قالب گرافی بدون جهت با لبه های برچسب خورده کد می شود. به طور کلی گراف همسایگی نواحی $(RAG-LE)$ طی مراحل زیر از شکل بخش بندی شده ایجاد می شود:

۱- یک نود به هر ناحیه نسبت داده می شود که حاوی برداری از اعداد حقیقی برای نمایش صفات ناحیه است.

۲- هر نود را به نواحی همسایه به وسیله لبه بدون جهت متصل می کنیم.

۳- بردار حقیقی به عنوان برچسب به هر لبه از گراف متصل می شود. این بردارارتباط مکانی دو ناحیه همسایه را توصیف می کند. به عنوان مثال در این آزمایش، به لبه بدون جهت بین نودهای i و j بردار $[D,A,B,C]$ نسبت داده می شود که:

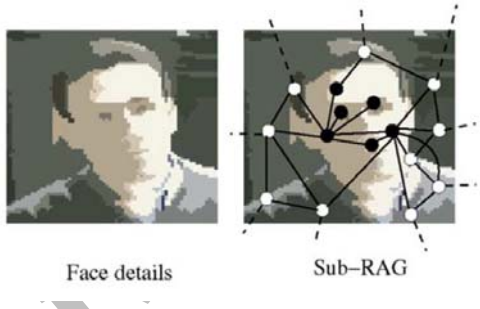
- D نمایش فاصله بین مرکز جرم دو ناحیه است،
- A نمایش زاویه بین محور اینرسی دو ناحیه است،
- B نمایش اندازه بین محور اینرسی ناحیه اول با خط واصل بین مرکز جرم ناحیه ها است،
- C اندازه زاویه بین محور اینرسی ناحیه دوم باخط واصل بین مرکز جرم ناحیه ها است.

در زمان آموزش مقدار هدف 1 به نودهایی که متعلق به قسمتی از شی مورد نظر هستند، نسبت داده می شود و مقدار هدف 0 به نواحی دیگر (خارج از شی) نسبت داده می شود. شکل ۵ $RAG-LE$ متناظر با تصویر نمونه را نشان می دهد. در این تصویر شی مورد نظر، تصویر صورت می باشد که نود متناظر با نواحی مربوط به

صورت، با رنگ سیاه (مقدار 1) مشخص شده اند. در حالی که نودهای نواحی اشیاء دیگر با رنگ سفید(مقدار 0) مشخص شده اند.

مدل RNN که در بخش قبل بحث شد، تنها $DAG-LE$ ها را پردازش می کند. بنا براین $RAG-LE$ حاصل از تصویر باید به گراف بدون چرخه جهت دار تبدیل شود. در این تبدیل نودی از گراف با نام n به عنوان ریشه درختی که قرار است ایجاد شود انتخاب می گردد. روش ایجاد درخت که در ادامه می آید برای تمام نودهای $RAG-LE$ یا حداقل نودهای رندوم تکرار می شود، تا نهایتا جنگلی از درخت ها که هم ارز بازگشتی با $RAG-LE$ است، ایجاد شود [۱۳].

قبل از تبدیل $RAG-LE$ به $DAG-LE$ ابتدا باید گراف غیر جهت دار به گراف جهت دار تبدیل شود. برای این کار یک زوج لبه با جهت عکس به جای لبه بدون جهت بین دو نود قرار می گیرد که هر لبه از این زوج همان اطلاعات لبه بدون جهت را دارا است. اگر برچسب $[D,A,B,C]$ توصیف شده در بالا متصل به لبه بین گره های i و j باشد، به لبه i به j برچسب $[D,A,B]$ و به لبه j به i برچسب $[D,A,C]$ نسبت داده می شود. گراف جهت دار به دست آمده، با روش زیر به درخت گسترش می یابد:



شکل ۵: $RAG-LE$ ایجاد شده روی تصویر بخش بندی شده

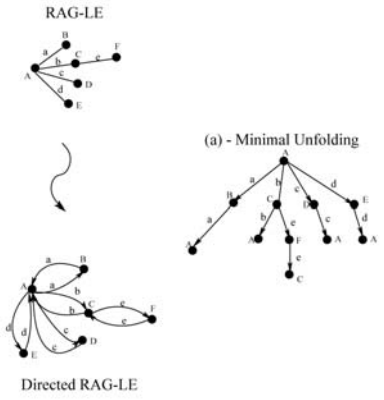
۱- یک کپی از n را در T قرار می دهیم.

۲- G را با شروع از n بروش عمقی پیمایش می کنیم. برای هر نود ملاقات شده v یک کپی از v به T اضافه می کنیم و v را به نود پدرش، برای حفظ اطلاعات لبه ها وصل می کنیم.

۳- مرحله (۲) را آنقدر تکرار می کنیم تا شرط از پیش تعیین شده محقق شود.

٤- مقدار هدف، 0 یا 1، را به ریشه t نسبت می دهیم.

شرط های زیادی برای اتمام مرحله (۲) وجود دارد. در هر شرطی که استفاده شود هر لبه باید حداقل یکبار ملاقات شود. درخت شکل (۶) با استفاده از همین مینیمم شرط باز شده است.



شکل ٦: تبدیل $RAG-LE$ به درخت هم ارزی بازگشت پذیر

۷ آزمایش

در این آزمایش تمرکز روی تشخیص صورت است. از ۵۰۰ تصویر ۳۸۴ صورت (هر تصویر شامل حداکثر یک صورت) استفاده شده است. هدف تشخیص مکان صورت می باشد. صورت ها در جهات و ابعاد مختلف هستند. تعداد ۲۰۰ تصویر برای آموزش و ۳۰۰ تصویر برای آزمایش استفاده شده است. ابتدا تصویر بخش بندی شده، $RAG-LE$ ایجاد می شود، برچسب نودها و لبه ها با اجزایی که در بخش قبل آمده محاسبه می شود. گراف به دست آمده به صورت جنگلی از درخت ها گسترش می یابد و شبکه RNN آموزش داده می شود تا مشخص کند ریشه هر درخت جزء صورت است یا نه. شبکه آموزش داده شده با مراحل زیر آزمایش می شود:

۱- تصویر بخش بندی شده و $RAG-LE$ ایجاد می شود.
۲- $RAG-LE$ به صورت جنگلی از درختان گسترش می یابد.
۳- هر درخت به وسیله شبکه RNN آموزش داده شده، پردازش می شود تا شبکه مشخص کند که آیا ریشه هر درخت متعلق به صورت است یاخیر.
۴- نواحی همسایه که جزئی از صورت هستند، ترکیب می شوند تا قاب دربرگیرنده مینیمم شامل صورت روی تصویر ایجاد شود.
۵- در صورتی تشخیص با موفقیت همراه است که مستطیل تشخیص داده شده مساوی مستطیل دربرگیرنده باشد.

برای مشخص شدن بهترین معماری، شبکه های عصبی بازگشت پذیر مختلف آموزش داده شده اند، که نتیجه آزمایش در جدول زیر آمده است. میزان دقت[8] از تقسیم تعداد نواحی درست تشخیص داده شده بر تمامی نواحی آزمایش شده به دست می آید و میزان صحت [9] از طریق صورت های درست تشخیص داده شده محاسبه می شوند. یک صورت به شرطی درست تشخیص داده می شود که برای قاب در برگیرنده صحیح متناظر و

[8] Accuracy rate
[9] Precision rate

پیش بینی شده، نسبت محاسبه شده از تقسیم اشتراک دو قاب بر اجتماع آن ها، کمتر از آستانه تعیین شده باشند (در این آزمایش آستانه ۹۰٪ در نظر گرفته شده است).

معماری *RNN*	میزان دقت *RNN*	میزان صحت
One layer – 5 state neurons	77.29%	78.23%
One layer – 10 state neurons	83.62%	85.33%
One layer – 15 state neurons	73.22%	72.18%

۸ نتیجه گیری

در این آزمایش روش جدیدی مبتنی بر نمود[10]، برای تشخیص اشیاء در تصاویر که با $DAG-LE$ ها نمایش داده می شوند، به کار گرفته شده است. این روش به دلیل استفاده از نمایش گرافیکی پایا در مقابل تبدیل و چرخش تصاویر پایاست و از قابلیت یادگیری مدل جدید شبکه عصبی بازگشت پذیر در تشخیص اجزا اشیاء استفاده می کند. این مدل جدید با $DAG-LE$ ها کار می کند در این روش نیاز به هیچ آگاهی قبلی از مدل شی مورد نظر نیست. نتایج آزمایش ها موثر بودن این روش را تایید می کند.

منابع

[1] M. Bianchini, M. Maggini, L. Sarti, and F. Scarselli, Recursive eural networks for processing graphs with labelled edges, in Proceedings of ESANN 2004, Bruges (Belgium), April 2004, pp. 325-330.

[2] M. Bianchini, M. Gori, and F. Scarselli, Recursive processing of cyclic graphs, IEEE International Joint Conference on Neural Networks, pp. 154-159, 2002.

[3] M. Bianchini, M. Gori, L. Sarti, and F. Scarselli, Backpropagation through cyclic structures, in LNAI - AI*IA 2003: Advanced in Artificial Intelligence, A. Cappelli and F. Turini, Eds., Pisa (Italy), September 2003, pp. 118-129, LNCS-Springer.

[4] M. Bianchini, M. gori and F. Scarselli, Theoretical properties of recursive networks with linear neurons, IEEE Trans. Neural networks, 953-967, 2001.

[5] P. Frasconi, M. Gori, and A. Sperduti, A general framework for adaptive processing of data structures, IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 768-786, September 1998.

[6] M. Gori, M. Maggini, and L. Sarti, A recursive neural network model for processing directed acyclic graphs with labeled edges, in Proceedings of the International Joint Conference on Neural Networks, 2003, pp. 1351-1355.

[7] B. Hammer and V. Sperschneider, Neural networks can approximate mapping on structured objects, in Int. Conf. Comput. Intell. Neural Networks P.P.Wang, Ed. 211-214, 1997.

[8] Approximation capabilities of folding networks, in ESANN, Bruges, Belgium, 33-38, 1999.

[9] A. Kuchler and C. Goller, Inductive learning in symbolic domains using structure-driven recurrent neural networks, in Advances in Artificial Intelligence, G. Gorz and S. Holldobler, Eds., pp. 183-197. Springer, Berlin, 1996.

[10] A. Kuchler, Adaptive processing of structured data: From sequence to trees and beyond, PH.D. Dissertation, Faculty comput. Sci. Univ. Ulm, 1999.

[11] L. Lenz, Group theoretical transforms in image Processing, Current topics pattern recognition Res, 83-106, 1994.

[12] Optimal filters for the detection of linear patterns in 2-D and higher dimensional images, Pattern Recognition 163-172, 1987.

[10] Appearance-based

[13] Y. Lecun, B. Boser, J. S. Denker, D.Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, Backpropagtion applied handwritten zip code recognition, Neural Comput, 541-551, 1989.
[14] Y. Lecun, L. Battou, Y. Bengio, and P. Haffner, Gradient based learning applied to document recognition, Proc. IEEE. 2278-2324, 1998.
[15] A. Sperduti and A. Starita, Supervised neural networks for the classification of structures, IEEE Transactions on Neural Networks, vol. 8, pp. 429-459, 1997.